

Tutorial :Sensor de Ultrassom (HC-SR04)

Propósito

O propósito de apresentar o Sensor de Ultrassom (HC-SR04) é para entender e aprender a usar este. Este sensor será utilizado para medir distâncias e detectar obstáculos. Existem muitas bibliotecas para este sensor (ex: arduino)mas aqui não serão usados para podermos entender o funcionamento deste.

Objetivo

Depois de completar este assunto, Sensor de Ultrassom (HC-SR04), será capaz de:

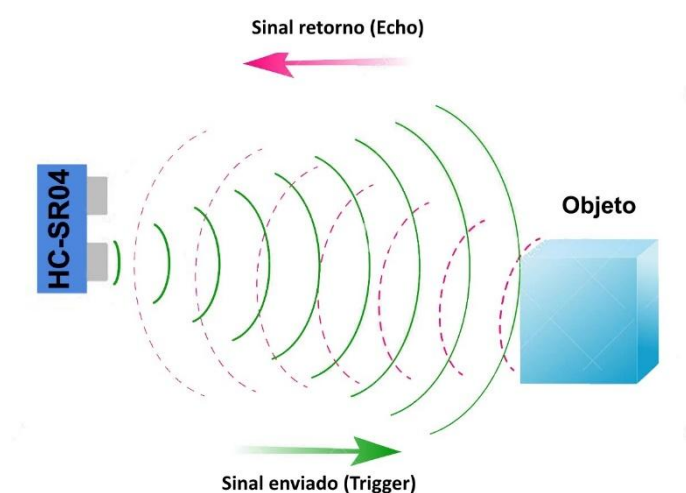
- Conhecer o Sensor de Ultrassom (HC-SR04)
- Utilizar e conectar o Sensor de Ultrassom (HC-SR04) a algum microcontrolador.
- Trabalhar com diferentes aplicações práticas para dominar este sensor.

Teoria

Apresentamos o **sensor de ultrassom** que serve para detectar obstáculos e medir distâncias e além disso tem inúmeras aplicações. Eles estão presentes nos famosos “sensores de estacionamento” dos automóveis mais modernos. Aqueles que apitam quando você está perto demais de uma parede, outro veículo ou obstáculo. Além de detectar a presença de uma barreira, eles também conseguem medir a distância do obstáculo. Por isso, além dos carros, também são muito usados em fábricas, processos de automação industrial, robótica e até na medicina.

Princípio de Funcionamento

Tudo começa pela emissão de um pequeno pulso sonoro de alta frequência que se propagará na velocidade do som do meio em questão. Quando este pulso atingir um objeto, um sinal de eco será refletido para o sensor. A distância entre o sensor e o objeto pode então ser calculada se soubermos o tempo entre a emissão e a recepção do sinal e a velocidade do som no meio em questão.



O **sensor de ultrassom** é composto de um emissor e um receptor de ondas sonoras. Ambos trabalham com ondas de altíssima frequência, na faixa dos **40.000 Hz** (ou 40KHz). Isto é muito, muito acima do que os nossos ouvidos são capazes de perceber. O ouvido humano consegue,

normalmente, perceber ondas entre 20 e 20.000 Hz e por isto o sinal emitido pelo sensor ultrassônico passa despercebido por nós.

O sinal emitido, ao colidir com qualquer obstáculo, é refletido de volta na direção do sensor. Durante todo o processo, o aparelho está com uma espécie de “cronômetro” de alta precisão funcionando. Assim, podemos saber quanto tempo o sinal levou desde a sua emissão até o seu retorno. Como a velocidade do som no ar é conhecida, é possível, de posse do tempo que o sinal levou para ir até o obstáculo e voltar, calcular a distância entre o sensor e o obstáculo. Para isto vamos considerar a velocidade do som no ar (340 m/s) na seguinte equação:

$$d = (V * t) / 2$$

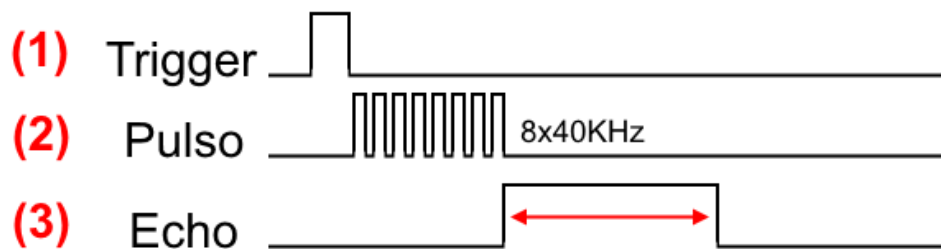
Onde:

d = Distância entre o sensor e o obstáculo (é o que queremos descobrir).

V = Velocidade do som no ar (340 m/s).

t = Tempo necessário para o sinal ir do sensor até o obstáculo e voltar (é o que o nosso módulo sensor ultrassom mede).

A divisão por dois existe já que o tempo medido pelo sensor é na realidade o tempo para ir e voltar, ou seja, duas vezes a distância que queremos descobrir.



Primeiramente é enviado um pulso de 10μs (Trigger), indicando o início da transmissão de dados. Depois disso, são enviados 8 pulsos de 40 KHz automaticamente e o sensor então aguarda o retorno (em nível alto/high), para determinar a distância entre o sensor e o objeto.

Detalhes do sensor de ultrassom

Aqui utilizaremos o sensor de ultrassom HC-SR04 que é bastante comum, tem boa precisão e acessível.



Estes módulos possuem quatro pinos e são bem simples de serem utilizados com vários tipos de microcontroladores como o Arduino. Dois pinos são utilizados para alimentar o sensor (**Vcc** e **GND**), um deles é utilizado para disparar o sinal ultrassônico (**Trig**) e o outro para medir o tempo que ele leva para retornar ao sensor (**Echo**).

Os pinos do sensor são os seguintes:

Vcc : Alimentação do módulo com +5 V.

Trig : Gatilho para disparar o pulso ultrassônico. Para disparar coloque o pino em HIGH por pelo menos 10µs.

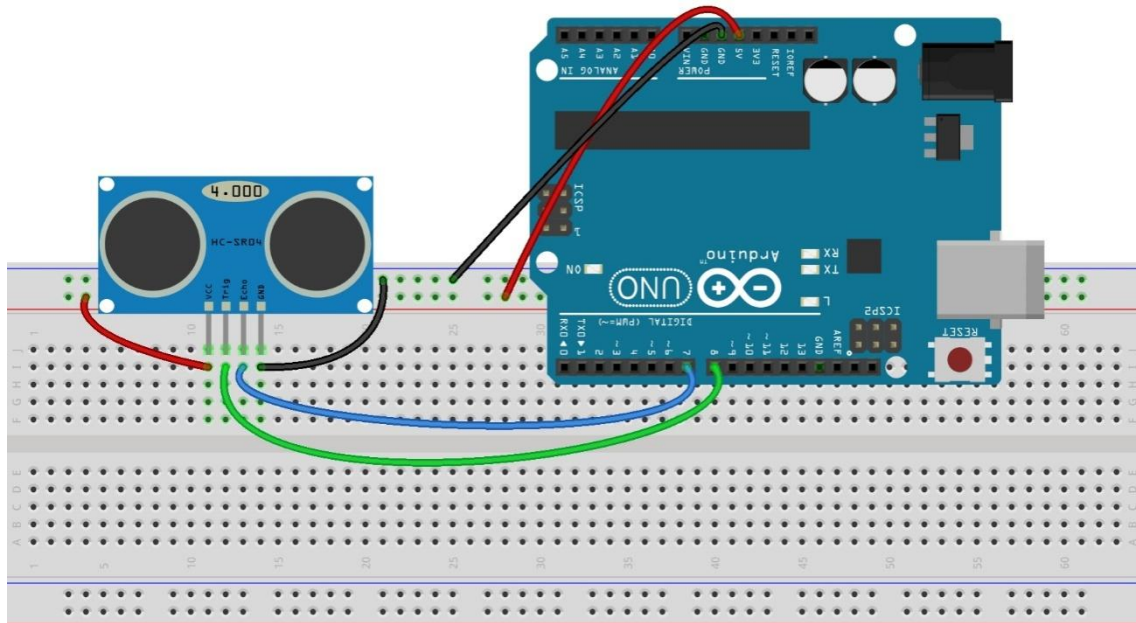
Echo : Gera um pulso com a duração do tempo necessário para o eco do pulso ser recebido pelo sensor.

GND : Terra.

Devemos levar em consideração que tipo de obstáculo estamos querendo detectar. Se o obstáculo for muito pequeno pode ser que ele não gere um sinal de retorno suficiente para ser percebido pelo sensor. Se o obstáculo não estiver posicionado bem na frente do sensor você pode ter medidas imprecisas ou até mesmo não acusar a presença do mesmo. E finalmente, a faixa de distância que o sensor trabalha fica entre 2cm e 4m e isto pode variar de sensor para sensor.

Exemplos do sensor de ultrassom com o arduino

Para este primeiro exemplo utilizaremos o seguinte circuito:



fritzing

```
/*  
EXEMPLO 1. Neste programa sera mostrado o tempo que demora em voltar o som quando bater em  
algum obstaculo. Lembre-se que o valor sera em microsegundos e apresentado no PC pela porta serial.  
*/  
const int echoPin = 7; //Pino 7 recebe o pulso do echo  
const int trigPin = 8; //Pino 8 envia o pulso para gerar o echo  
void setup()  
{  
  Serial.begin(9600); //inicia a porta serial  
  pinMode(echoPin, INPUT); // define o pino 7 como entrada (recebe)  
  pinMode(trigPin, OUTPUT); // define o pino 8 como saída (envia)  
}  
void loop()  
{  
  //seta o pino 8 com um pulso baixo "LOW" ou desligado ou ainda 0  
  digitalWrite(trigPin, LOW);  
  // atraso de 2 microssegundos  
  delayMicroseconds(2);  
  //seta o pino 8 com pulso alto "HIGH" ou ligado ou ainda 1  
  digitalWrite(trigPin, HIGH);  
  //atraso de 10 microssegundos  
  delayMicroseconds(10);  
  //seta o pino 8 com pulso baixo novamente  
  digitalWrite(trigPin, LOW);  
  //a função pulseIn eh o tempo de permanencia em high do pino echoPin.  
  long tempo = pulseIn(echoPin, HIGH);  
  Serial.print("Tempo: ");  
  Serial.print(tempo);  
  Serial.println(" useg");  
  delay(1000); //espera 1 segundo para fazer a leitura novamente  
}
```

Exemplo 2: Utilizando o circuito anterior vamos a determinar a distância de algum objeto na frente do sensor de ultrassom. Esta distância será mostrada no PC. Os valores que serão apresentados estarão em centímetros.

```
/*
EXEMPLO 2. Neste programa seradeterminada a distancia de algum objeto na frente do sensor de
ultrasom. Esta distancia estara em centímetros e sera apresentada no PCpela porta serial.
*/
constintechoPin = 7; //Pino 7 recebe o pulso do echo
constinttrigPin = 8; //Pino 8 envia o pulso para gerar o echo
void setup()
{
  Serial.begin(9600); //inicia a porta serial
  pinMode(echoPin, INPUT); // define o pino 7 como entrada (recebe)
  pinMode(trigPin, OUTPUT); // define o pino 8 como saída (envia)
}
void loop()
{
  //seta o pino 8 com um pulso baixo "LOW" ou desligado ou ainda 0
  digitalWrite(trigPin, LOW);
  // atraso de 2 microssegundos
  delayMicroseconds(2);
  //seta o pino 8 com pulso alto "HIGH" ou ligado ou ainda 1
  digitalWrite(trigPin, HIGH);
  //atraso de 10 microssegundos
  delayMicroseconds(10);
  //seta o pino 8 com pulso baixo novamente
  digitalWrite(trigPin, LOW);
  //a função pulseIn eh o tempo de permanencia em high do pino echoPin.
  float tempo = pulseIn(echoPin,HIGH);

  //Esse calculo eh baseado em  $s = v \cdot t$ , lembrando que o tempo vem dobrado
  //porque eh o tempo de ida e volta do ultrassom
  float distancia = tempo /29.4 / 2 ;

  Serial.print("Distancia em cm: ");
  Serial.println(distancia);
  delay(1000); //espera 1 segundo para fazer a leitura novamente
}
```

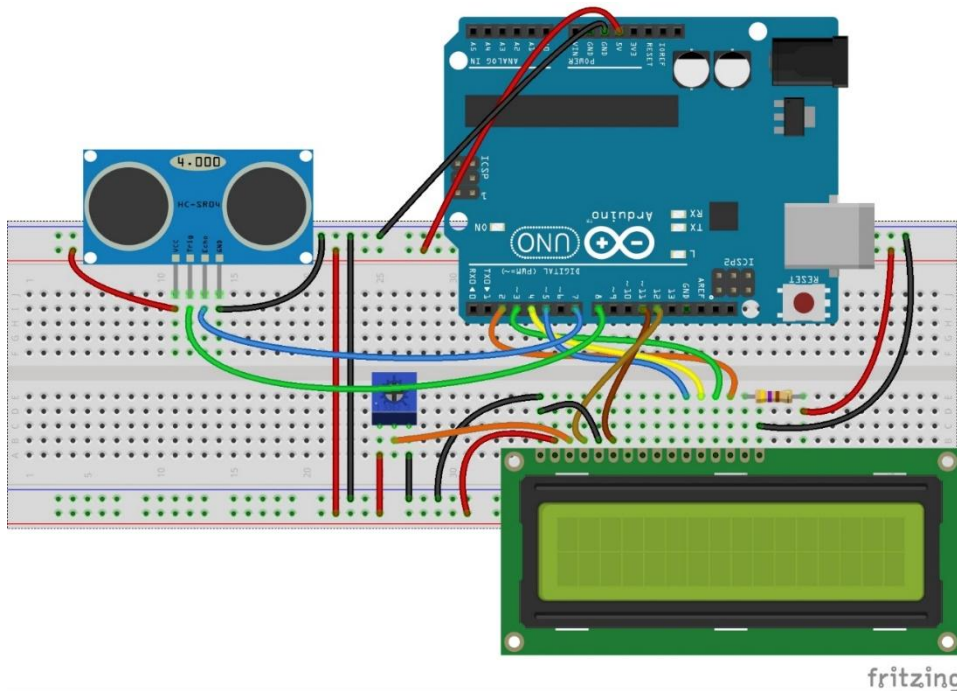
O código é bastante simples e auto-explicativo, mas você deve estar se perguntando de onde veio o número 29.4 no cálculo da distância, correto? Como queremos saber a distância, em centímetros, tivemos que fazer algumas continhas. Primeiro calculamos a velocidade do som no ar em centímetros por segundo. Isto deu: 34000cm/s. Depois calculamos quantos microssegundos o som leva para percorrer um centímetro. Para isto basta resolver a seguinte regra de três:

1000000us=> 34000 cm

X us => 1 cm

E temos que são necessários 29.4 us para o som percorrer um centímetro. Pronto, agora como medimos o tempo em microssegundos para o som ir e voltar ao sensor, basta dividir este tempo por 29.4 para descobrir quantos centímetros o som teve que percorrer. Porém, como o som tem que fazer o trajeto de ida e volta, finalizamos dividindo o resultado por dois para sabermos a distância entre o sensor e o obstáculo.

Exemplo 3: Ao circuito anterior vamos acrescentar um LCD (ver figura) onde será mostrada a distância de algum objeto na frente do sensor de ultrassom. Os valores que serão apresentados estarão em centímetros.



```

/*
EXEMPLO 3. Neste programa sera determinada a distancia de algum objeto na frente do sensor de
ultrassom. Esta distancia estara em centímetros e sera apresentada no LCD.
*/
#include <LiquidCrystal.h> // incluimos o código da biblioteca

// constantes para o número de linhas e colunas no LCD
const int numLinhas = 2;
const int numCols = 16;

// inicializar a biblioteca do LCD com números dos pinos de interface
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Pinos do sensor de ultrassom
const int echoPin = 7; // Pino 7 recebe o pulso do echo
const int trigPin = 8; // Pino 8 envia o pulso para gerar o echo

void setup()
{
  lcd.begin(numCols, numLinhas);
  pinMode(echoPin, INPUT); // define o pino 7 como entrada (recebe)
  pinMode(trigPin, OUTPUT); // define o pino 8 como saída (envia)
  lcd.setCursor(0,0); // colocando o cursor na coluna 0 e na linha 1
  lcd.print("Distancia em cm: ");
}

void loop()
{
  // seta o pino 8 com um pulso baixo "LOW" ou desligado ou ainda 0
  digitalWrite(trigPin, LOW);
  // atraso de 2 microssegundos
  delayMicroseconds(2);

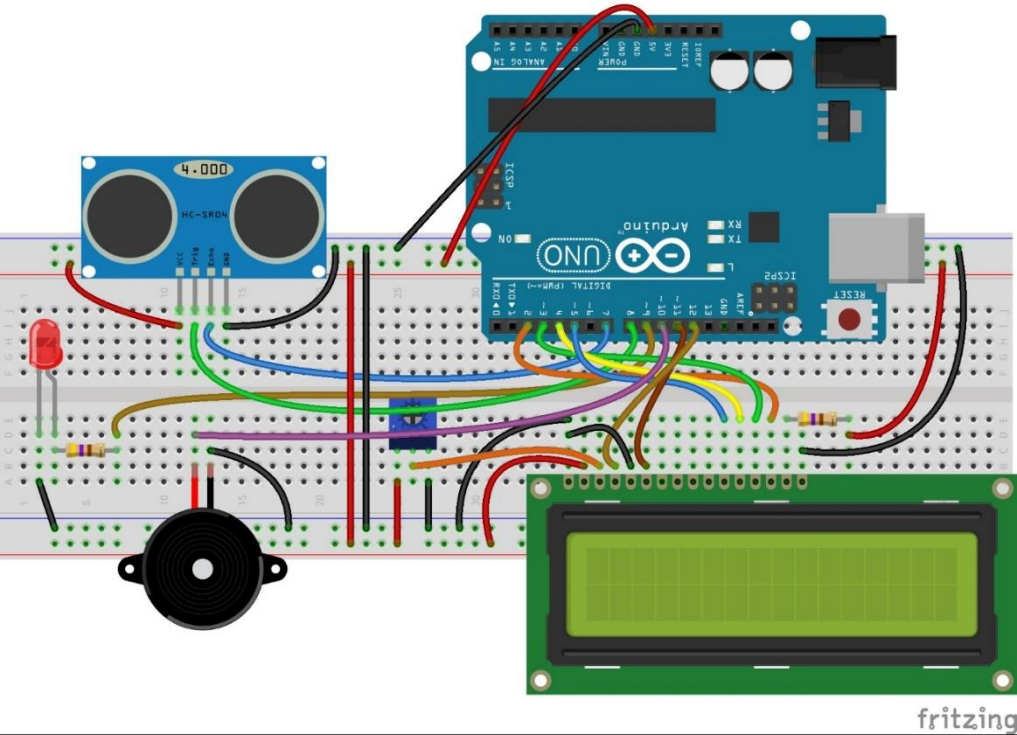
```

```
//seta o pino 8 com pulso alto "HIGH" ou ligado ou ainda 1
digitalWrite(trigPin, HIGH);
//atraso de 10 microssegundos
delayMicroseconds(10);
//seta o pino 8 com pulso baixo novamente
digitalWrite(trigPin, LOW);
//a função pulseIn eh o tempo de permanencia em high do pino echoPin.
float tempo = pulseIn(echoPin,HIGH);
//Esse calculo eh baseado em  $s = v \cdot t$ , lembrando que o tempo vem dobrado
//porque eh o tempo de ida e volta do ultrassom
float distancia = tempo / 29.4 / 2 ;
lcd.setCursor(0,1); //colocando o cursor na coluna 0 e na linha 2
lcd.print(distancia);
delay(1000); //espera 1 segundo para fazer a leitura novamente
}
```

Exemplo 4: Ao circuito do exemplo 3 acrescentamos um led, um buzzer e vamos fazer que ele funcione de acordo com a tabela mostrada:

Distância do objeto ao sensor	Pisca (led) e apita(buzzer)
Acima de 1 m	Fica desligado completamente
Entre 80cm e 1 m	Cada 1 seg
Entre 60 cm e 80 cm	Cada 0,75 seg (750 milisegundos)
Entre 40 cm e 60 cm	Cada 0,5 seg (500 milisegundos)
Entre 20 cm e 40 cm	Cada 0,3 seg (300 milisegundos)
menor que 20 cm	Fica ligado completamente

As distâncias devem ser mostradas no LCD.



```
/*
EXEMPLO 4. Neste programa sera determinada a distancia de algum objeto na frente do sensor de
ultrasom, a distancia sera mostrada no LCD. So que este sistema trabalhara de acordo com a seguinte
tabela:

=====
Distância do objeto ao sensor      Pisca (led) e apita(buzzer)
=====
Acima de 1 m                       Fica desligado completamente
Entre 80cm e 1 m                   Cada 1 seg
Entre 60 cm e 80 cm                Cada 0,75 seg (750 milisegundos)
Entre 40 cm e 60 cm                Cada 0,5 seg (500 milisegundos)
Entre 20 cm e 40 cm                Cada 0,3 seg (300 milisegundos)
menor que 20 cm                    Fica ligado completamente

*/
#include<LiquidCrystal.h> // incluimos o codigo da bilbioteca
//constantes para o numero de filas e colunas no LCD
const int numLinhas = 2;
const int numCols = 16;
```



```

// inicializar a biblioteca do LCD com numeros dos pinos de interface
LiquidCrystallcd(12, 11, 5, 4, 3, 2);

//Pinos do sensor de ultrasom
constintechoPin = 7; //Pino 7 recebe o pulso do echo
constinttrigPin = 8; //Pino 8 envia o pulso para gerar o echo

constintled = 9; //pino 9 ligado ao led
constintbuzz = 10; //pino 10 ligado ao buzzer

void setup()
{
  pinMode(echoPin, INPUT); // define o pino 7 como entrada (recebe)
  pinMode(trigPin, OUTPUT); // define o pino 8 como saída (envia)

  pinMode(led, OUTPUT); //define o pino 9 como saída
  pinMode(buzz, OUTPUT); //define o pino 10 como saída

  lcd.begin(numCols, numLinhas);
  lcd.setCursor(0,0); //colocando o cursor na coluna 0 e na linha 1
  lcd.print("Distancia em cm: ");
}

void loop()
{
  //seta o pino 8 com um pulso baixo "LOW" ou desligado ou ainda 0
  digitalWrite(trigPin, LOW);
  // atraso de 2 microssegundos
  delayMicroseconds(2);
  //seta o pino 8 com pulso alto "HIGH" ou ligado ou ainda 1
  digitalWrite(trigPin, HIGH);
  //atraso de 10 microssegundos
  delayMicroseconds(10);
  //seta o pino 8 com pulso baixo novamente
  digitalWrite(trigPin, LOW);
  //a função pulseIn eh o tempo de permanencia em high do pino echoPin.
  float tempo = pulseIn(echoPin,HIGH);

  //Esse calculo eh baseado em  $s = v \cdot t$ , lembrando que o tempo vem dobrado
  //porque eh o tempo de ida e volta do ultrassom
  float distancia = tempo /29.4 / 2 ;
  lcd.setCursor(0,1); //colocando o cursor na coluna 0 e na linha 2
  lcd.print(distancia);
  if(distancia> 100)
  {
    digitalWrite(led,LOW);
    digitalWrite(buzz,LOW);
  }
  elseif(distancia >= 80 && distancia < 100)
  {
    digitalWrite(led,HIGH);
    digitalWrite(buzz,HIGH);
    delay(1000);
    digitalWrite(led,LOW);
    digitalWrite(buzz,LOW);
    delay(1000);
  }
  else if(distancia>= 60 && distancia< 80)
  {
    digitalWrite(led,HIGH);
    digitalWrite(buzz,HIGH);
  }
}

```

```
delay(750);
digitalWrite(led,LOW);
digitalWrite(buzz,LOW);
delay(750);
}
else if(distancia>= 40 &&distancia< 60)
{
digitalWrite(led,HIGH);
digitalWrite(buzz,HIGH);
delay(500);
digitalWrite(led,LOW);
digitalWrite(buzz,LOW);
delay(500);
}
else if(distancia>= 20 &&distancia< 40)
{
digitalWrite(led,HIGH);
digitalWrite(buzz,HIGH);
delay(300);
digitalWrite(led,LOW);
digitalWrite(buzz,LOW);
delay(300);
}
else
{
digitalWrite(led,HIGH);
digitalWrite(buzz,HIGH);
}
}
```

Exercícios:

1. Vamos colocar 5leds (segundo a figura) os quais ligaram da seguinte forma a depender da distância do obstáculo na frente do sensor de ultrassom:

Números de leds	Distância
1led	Acima de 1 m
2leds	Entre 70 cm e 1 m
3leds	Entre 40 cm e 69 cm
4leds	Entre 20cm e 39 cm
5leds	Menos que 20 cm

